JOURNAL OF AIRCRAFT
Vol. 45, No. 6, November–December 2008

Air Data Computation Using Neural Networks

Alberto Calia,* Eugenio Denti,† Roberto Galatolo,‡ and Francesco Schettini§
University of Pisa, 56122 Pisa, Italy

DOI: 10.2514/1.37334

The paper deals with the use of neural networks for the determination of pressure altitude and Mach number of a fly-by-wire high-performance aircraft during flight. In previous works the authors developed a methodology based on polynomial calibration functions for the determination of such flight parameters, together with the angles of attack and sideslip. Such an approach provided successful results, but the use of different polynomial functions in different areas was needed to map the entire flight envelope. The fading methodologies for the management of polynomial functions overlap and considerably increased both procedure complexity and the time to spent for the procedure tuning. In particular, the calibration functions related to the Mach number and static-pressure estimation are susceptible to these problems because of their high nonlinearity. The alternative approach studied in this paper, based on neural networks, provides a level of accuracy comparable with that of polynomial functions. However, such an approach is simpler, because it allows the entire flight envelope to be mapped by means of a single network for each output parameter, and so it eliminates the fading problems. In addition, the new procedure is extremely easier to tune when new data from flight tests are available. This is a very important point, because several versions of the air data computation algorithms are generally to be developed in parallel with the flight-envelope enlargement of a new aircraft.

Nomenclature

M_∞	=	asymptotic Mach number
$M_{\infty i}$	=	Mach number estimate from the ith probe
$P_{\mathrm{front}i}$	=	frontal pressure measured by the ith probe
P_{sa}	=	asymptotic static pressure
$P_{\mathrm{sa}i}$	=	static-pressure estimate from the ith probe
$P_{\mathrm{slot}i}$	=	slot pressure measured by the ith probe
α	=	angle of attack
β	=	angle of sideslip
λ_i	=	local flow angle of the ith probe

I. Introduction

THE air data system consists of all the elements that allow the flight parameters (static pressure, Mach number, and angles of attack and sideslip) to be evaluated during flight. This is obtained by properly processing the local airflow measurements provided by external air data probes. The air data system studied in this paper refers to that designed for the new jet trainer Alenia Aermacchi M-346, which employs four self-aligning air data probes, called integrated multifunction probes [1], installed on the fuselage (Fig. 1).

The shape of the probes is a truncated cone, with the axis normal to the surface of the fuselage. Each probe has five pressure slots distributed at equal intervals at an angle of 180 deg (Fig. 2). Two of the five slots (at 45 and −45 deg with respect to the central one) allow the probe to be aligned according to the direction of the local flow ([1,2]) by means of a vane mechanism, which rotates the probe until the pressures at the two slots are equal. The other three slots are devoted to local airflow pressure measurements. Each probe provides three outputs: the local flow angle λ_i, measured by a rotary transducer (where subscript $i = 1, \ldots, 4$ refers to the probe number); the frontal pressure $P_{\mathrm{front}i}$ ("like-total pressure"), provided by the frontal slot, aligned with the local flow direction, and the slot pressure $P_{\mathrm{slot}i}$ ("like-static pressure"). The last is obtained as the average of the pressures measured by the two slots at 90 deg from the local flow direction. In previous works [3–5] the authors developed computation algorithms that were able to determine the aforementioned flight parameters on the basis of the 12 signals coming from the four probes, including a proper management of system redundancy. Such algorithms use least-squares polynomial functions to approximate a database coming from wind-tunnel tests, and they take into account the effects of aircraft maneuvers and aircraft configuration. Successful results were obtained with this approach, even if it was necessary to use different polynomial functions in different areas of the flight envelope to guarantee the requested accuracy. As a consequence, the development of proper fading capabilities was also necessary, to manage the transition from different polynomial functions at the boundary areas. This problem suggested the opportunity of investigating alternative approaches such as the one based on neural networks.

The research described in this work is focused on the application of neural networks to the computation of static pressure P_{sa} and Mach number M_∞, because the calibration functions of such parameters are characterized by the higher nonlinearities and they are more difficult to fit with polynomial functions. However, the authors are carrying out research on the application of the neural networks to the estimation of the angles of attack and sideslip, but the results are not yet available at this time.

In conclusion, two different neural networks are developed and tuned in this paper for each probe: one for the evaluation of P_{sa} and one for M_∞. The management of system redundancy (monitoring and voting on the four estimates of P_{sa} and M_∞) is that described in [3–5] and it does not differ when using the neural network or the polynomial functions. The comparison between the two approaches was carried out with reference to the cruise configuration of the aircraft.

II. Neural Networks Architecture

An artificial neural network consists of a number of interconnected processing elements called neurons. Each input to the neurons (Fig. 3) has a corresponding weight w_k. The total input i_{total} is obtained by summing the products of the inputs and the weights. This

Received 28 February 2008; revision received 30 May 2008; accepted for publication 30 May 2008.

*Aerospace Systems Engineer, Department of Aerospace Engineering; a.calia@ing.unipi.it.

†Associate Professor of Flight Dynamics, Department of Aerospace Engineering; e.denti@ing.unipi.it.

‡Associate Professor of Aerospace Systems, Department of Aerospace Engineering; r.galatolo@ing.unipi.it.

§Research Engineer of Flight Mechanics, Department of Aerospace Engineering; f.schettini@ing.unipi.it.

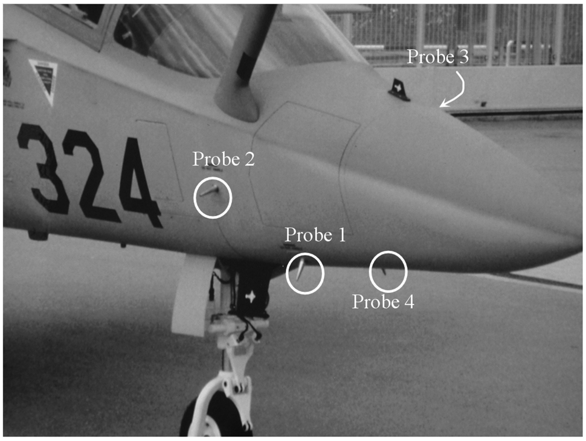

Fig. 1 Probe installation on Aermacchi M-346.

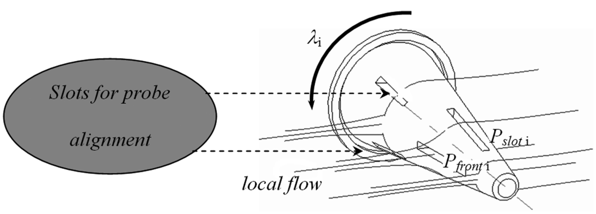

Fig. 2 Sketch of a single probe.

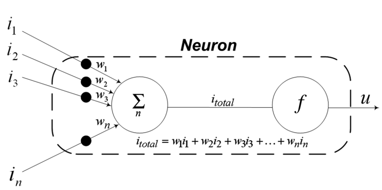

Fig. 3 Single neuron element.

signal is then passed through an activation function f, which is typically nonlinear [6].

The activation function used in this work is the hyperbolic tangent sigmoid function [7] that is fairly common in neural network applications:

$$u = \frac{2}{1 + \exp\left[-2\left(\sum_{k=1}^{n} w_k i_k\right)\right]} - 1 \tag{1}$$

The general neural network architecture chosen is a standard multilayer [6] constituted of three layers of neurons (Fig. 4): an input layer with p neurons directly connected with the input signals, a hidden layer in which m neurons have no external inputs, and an output layer in which j neurons produce the final network outputs.

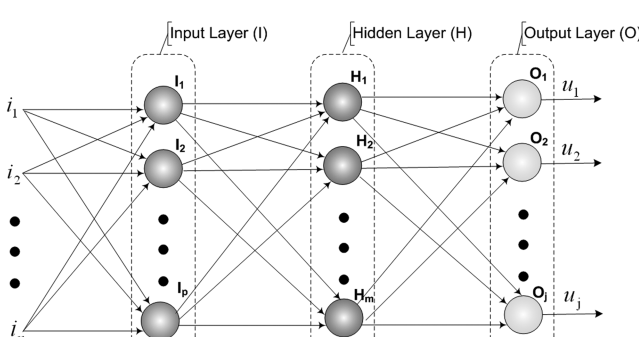

Fig. 4 Example of a neural network.

The optimal number of neurons in the hidden layer is a specific problem and depends on the fidelity of the mapping required and on the availability of training data. The use of too many hidden-layer neurons leads to an overfitting of the training data and, subsequently, to poor generalization properties of the network. In addition, the computation time required by the training process increases. If the number of hidden-layer neurons is insufficient, the network is not able to represent the complexity of the physical problem to be solved.

III. Computation of Static Pressure and Mach Number

The computation of static pressure and Mach number is performed by two independent networks for each probe. There are no deterministic means to determine the best network architecture; consequently, the proposed architecture was optimized heuristically during the early stages of development. The trial-and-error search led to two networks with a similar structure (Fig. 5).

The network relevant to the estimation of Mach number has three input signals and three layers of neurons: an input layer and a hidden layer of 20 neurons each and an output layer of a single neuron. Such an architecture needs 521 parameters to be stored in flight control computers (FCCs) for each probe. The output neuron provides the estimate of the Mach number $M_{\infty i}$, and the three input signals are the angles of attack α and sideslip β and the ratio $P_{\text{front}i}/P_{\text{slot}i}$ between the pressures measured by the ith probe. This pressure ratio provides important information to the network because it is highly dependent on the Mach number. To give an example of such a dependency, in Fig. 6, the pressure ratio measured by probe 2 within the wind-tunnel tests is plotted against α and β for various Mach numbers. The figure

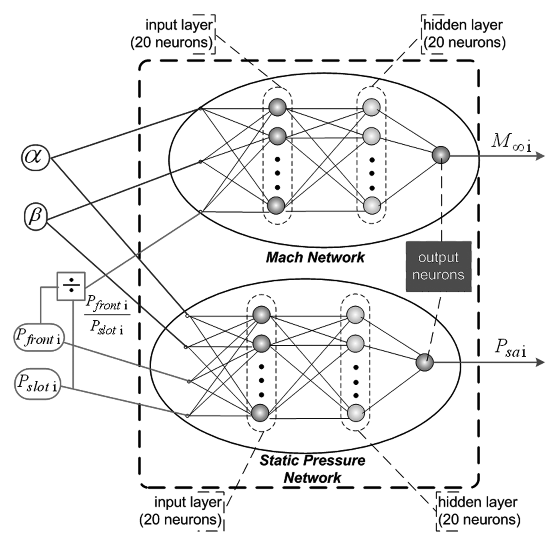

Fig. 5 Mach and static-pressure networks (related to the *i*th probe).

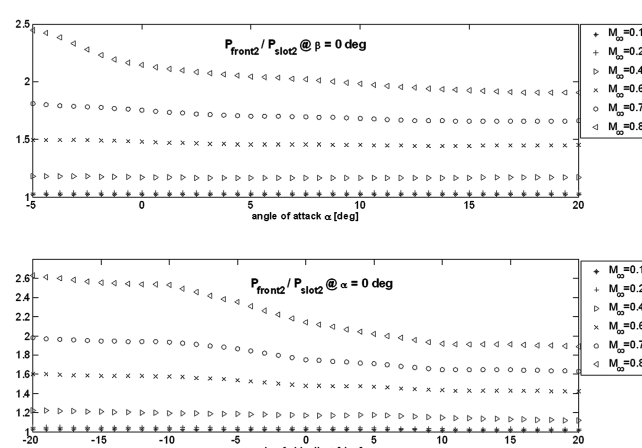

Fig. 6 Pressure ratio vs α, β, and Mach (probe 2).

also outlines some effects of α and β on the pressure measurements. The inputs α and β to the network aim to compensate for such effects.

The network for the estimation of static pressure $P_{\mathrm{sa}i}$ has the same architecture (Fig. 5), except for the input data that are, in this case, the angles α and β and the pressures $P_{\mathrm{front}i}$ and $P_{\mathrm{slot}i}$ measured by the ith probe. The static-pressure network needs 541 parameters for each probe.

The eight networks (two for each probe) need a total of $(521 + 541) \times 4 = 4248$ parameters to be stored in the FCCs. This number of parameters has to be compared with the approximately 1200 coefficients needed by using the polynomial functions [3] for the computation of pressures for the cruise-aircraft configuration (no deflection of leading-edge and trailing-edge flaps). However, the advantages of the neural networks will be outlined in the Conclusions.

The neural networks were trained in two ways: by using wind-tunnel-test data and by using preliminary flight-test data. The training based on wind-tunnel data was carried out to compare the accuracies of the neural networks with those of the method developed in [3] based on polynomial functions. The training based on flight-test data was carried out to examine the ability of the neural networks to be tuned on actual flight data and to determine the possible improvement of the accuracy that can be achieved.

IV. Training Process with Wind-Tunnel Database

A set of training data was obtained from a database provided by Alenia Aermacchi relevant to the new jet trainer Aermacchi M-346. The data refer to conditions of rectilinear motions and to a large part of the flight envelope, in terms of angles of attack and sideslip and Mach number.

The neural networks are excellent tools for interpolating a given set of data if the training process uses a subset of points that properly represent the entire operating domain. The original database comprises approximately 50,000 points uniformly distributed over the four-dimensional space $(\alpha, \beta, P_{\mathrm{sa}}, M_\infty)$. The networks were trained on a subset of such data, and the networks' abilities to generalize different data from the training set were analyzed using a larger, complementary, subset of the same data.

The computation time needed for network training increases with the number of points used. In this case, the best results were obtained by using 1700 training points (Fig. 7), which were split in the two ranges of flight conditions shown in Table 1. Each of the ranges contains 850 points, uniformly distributed over the range itself, thanks to a random extraction from the wind-tunnel-data lookup tables. Such a division allowed a higher density of training points in the low-α range to obtain better performance over such a range.

The training was carried out by the Levenberg–Marquardt method, which belongs to the class of backpropagation error methods [7]. Such an iterative method finds a set of network weights by minimizing the mean square error. The error is calculated by comparing the predicted outputs with the target outputs. Figure 8 shows the mean square error of the predictions as a function of the number of the epochs relative to the training process performed on one of the neural networks relevant to the estimation of Mach number. The approximately 800 training epochs needed to minimize the error require approximately 15 min of computation time on a 3 GHz Pentium PC for each of the networks.

Table 1 Flight conditions considered for training points

	α, deg	β, deg	P_{sa}, Pa	M_∞
Low-α range	−5–5	−20–20	18,000–101,325	0.15–0.7
High-α range	5–25	−20–20	18,000–101,325	0.15–0.7

V. Training Process with Flight-Test Data

Alenia Aermacchi provided 20 flight time histories obtained on different days for different operative conditions of angle of attack, angle of sideslip, Mach number, and altitude and different maneuvers. During such flight tests the local flow angles and pressures measured by the four probes were recorded, together with the flight parameters to be used for the calibration of the air data system. The latter were measured by a nose boom installed on the aircraft.

The flight-test data comprise 7×10^5 points within the following flight envelope: $\alpha = 0$–20 deg, $\beta = -10$–10 deg, and $P_{\mathrm{sa}} = 20,000$–$100,000$ Pa, and $M_\infty = 0.2$–0.8.

The training subset was determined by means of the methodology described in [8], which allows a subset of uniformly distributed data to be extracted from a nonuniform data base. This is obtained by dividing the flight envelope into a number of equal bins. The size of each bin is $\Delta\alpha = 2$ deg, $\Delta\beta = 5$ deg, $\Delta P_{\mathrm{sa}} = 1200$ Pa, and $\Delta M_\infty = 0.01$.

As an example, Fig. 9 qualitatively shows the bins used for the extraction of the training data for the Mach number network. In this case, the bins are three-dimensional $(\alpha, \beta, M_\infty)$ because it is assumed that the static pressure does not affect the ratio $P_{\mathrm{front}i}/P_{\mathrm{slot}i}$, which is an input of this net (Fig. 5). In contrast, for the static-pressure network, the single bin is four-dimensional because the pressure P_{sa} affects the network inputs $P_{\mathrm{front}i}$ and $P_{\mathrm{slot}i}$.

From each bin, an equal number of training points was randomly extracted to have a uniform distribution within the bin itself. Such a number was chosen to obtain the total number of training points of the same order of magnitude as that used for the training based on the wind-tunnel-test data. It is worth noting that some bins contain a lower number of data, and some do not contain data at all.

This technique allowed approximately 1600 training points to be found, uniformly distributed into the three-dimensional domain

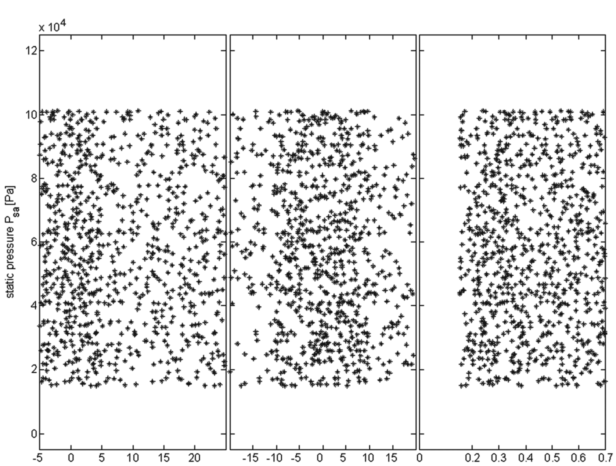

Fig. 7 Training points coming from wind-tunnel tests.

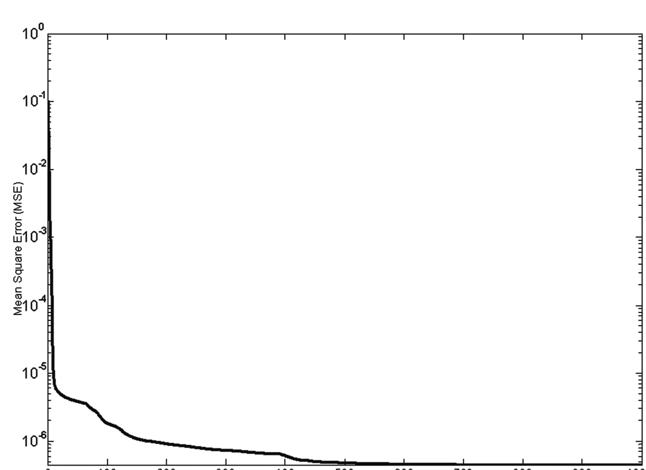

Fig. 8 Mean square error during training epochs (probe 2).

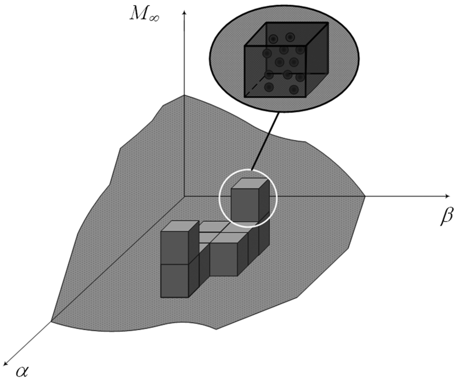

Fig. 9 Three-dimensional bin for training of the Mach network (flight-test points).

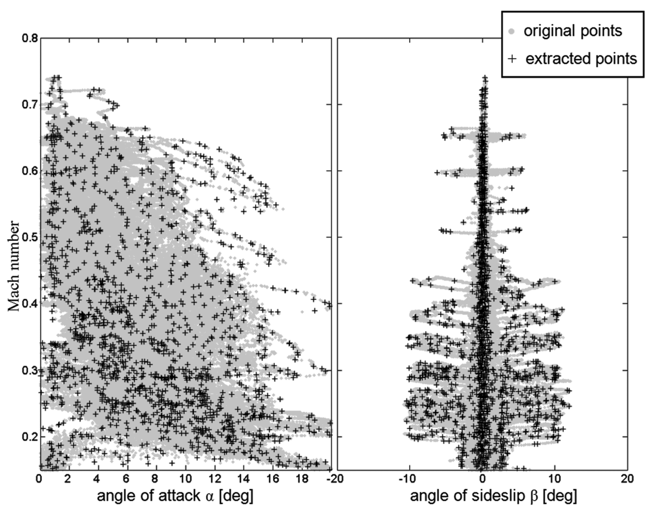

Fig. 10 Training points compared with original flight-test points.

$(\alpha, \beta, M_\infty)$, and 1600 uniformly distributed into the four-dimensional domain $(\alpha, \beta, M_\infty, P_{sa})$. Figure 10 shows the original and extracted points in the three-dimensional domain.

Finally, an analysis to verify the neural network sensitivity to the number of training points was performed. The results of such a study outlined that the use of 2000, 2500, or 3000 training points does not imply an evident improvement of a neural network's accuracy.

VI. Results

A. Networks Trained on Wind-Tunnel Data

The Mach and static-pressure networks were implemented in the MATLAB/Simulink® environment and extensively tested to assess their performance. For this purpose, the networks trained on wind-tunnel tests were interfaced with a flight simulator, which includes a model of integrated multifunction probes, based on the lookup tables coming from the same wind-tunnel tests. In this model, the probe measurement errors were modeled with random signals characterized as shown in Table 2. Concerning the probe dynamic model, the local-flow-angle response is provided by a first-order transfer function with a time constant equal to 0.054 s. The pressures transducers dynamics are faster and they have not been considered.

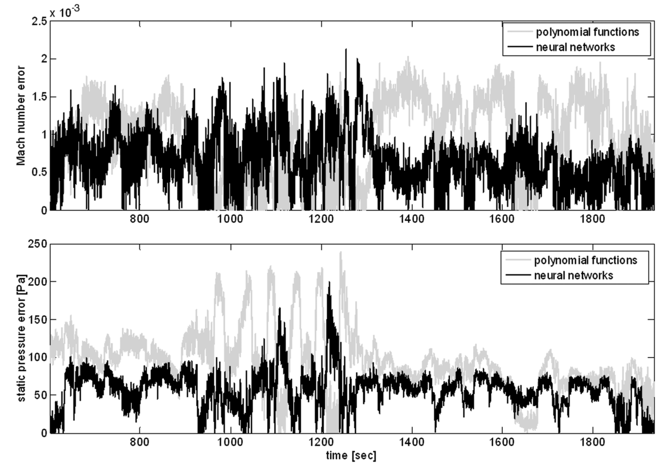

Fig. 11 Errors in Mach number and static pressure (maneuver simulation with a low angle of sideslip).

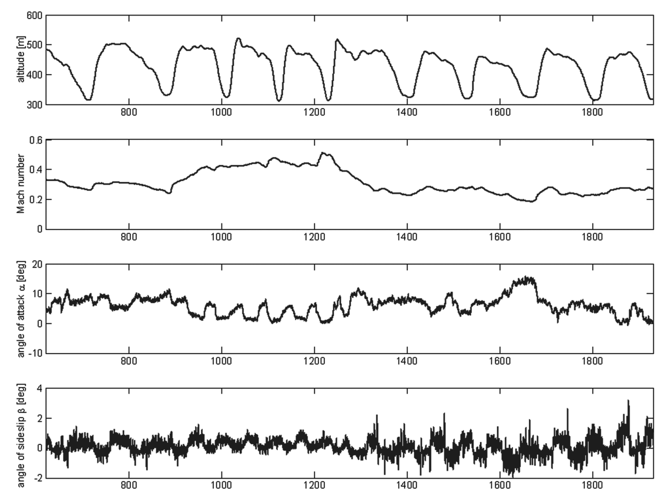

Fig. 12 Errors in Mach number and static pressure (maneuver simulation with a low angle of sideslip).

Both the angle measurements and pressure measurements are updated at a frequency of 80 Hz.

As an example, Fig. 11 reports the absolute values of Mach number and static-pressure errors provided by the two approaches (polynomial functions and neural networks) during a generic maneuver, generated with the flight simulator. The errors refer to the voted values calculated in both approaches by means of the voting algorithms described in [5]. During such a maneuver, the angle of attack, the Mach number, and the altitude varied within the ranges 0–20 deg, 0.2–0.6, and 300–500 m, respectively, whereas the variation of the angle of sideslip was very small (see Fig. 12). Figure 13 shows an analogous example in which the angle of sideslip varied within ± 10 deg. Concerning this maneuver, Fig. 14 shows the time histories of the altitude, Mach number, angle of attack, and angle of sideslip. In both cases, the neural networks produced better results than the polynomial functions, tuned on the same wind-tunnel data, for both the Mach number and the static-pressure computation. The maximum and mean errors and standard deviation of the neural networks and polynomial functions related to the two tests are shown in Table 3 and analogous results were found for all the other tests

Table 2 Range and standard deviation of the probe model

	λ_i	$P_{\text{front}i}$	$P_{\text{slot}i}$
Range	−50–50 deg	5000–310,000, Pa	2700–112,000, Pa
Standard deviation (% of full scale)	0.12%	0.002%	0.018%

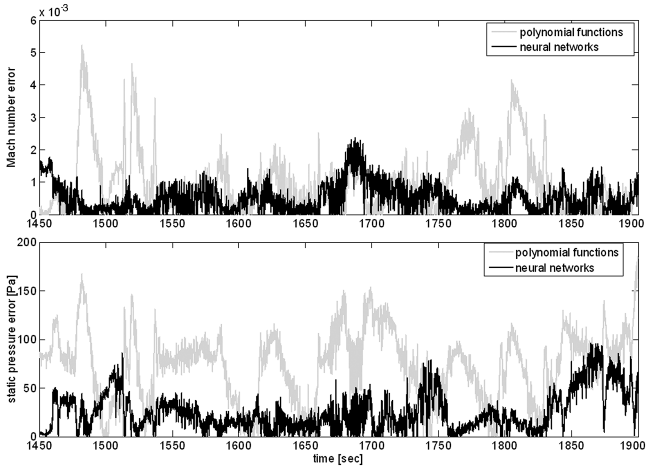

Fig. 13 Errors in Mach number and static pressure (maneuver simulation with $-10 < \beta < 10$ deg).

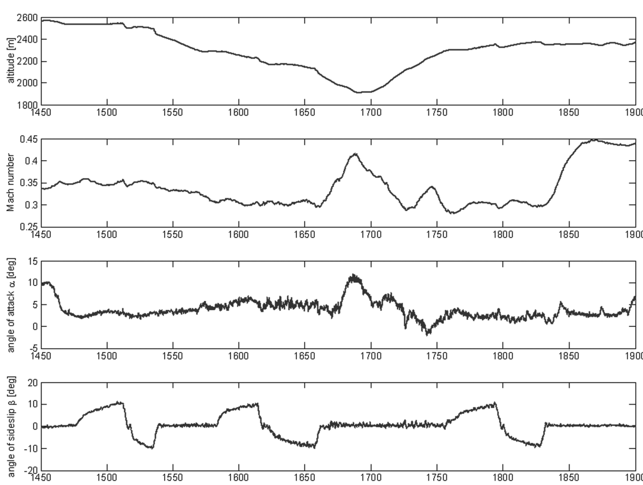

Fig. 14 Time histories for a maneuver with $-10 < \beta < 10$ deg.

performed. Such performances were obtained by using the nominal values of the angles of attack and sideslip as signal inputs for both the neural networks and the polynomial functions.

B. Networks Trained on Flight-Test Data

In this case, the assessment of the neural networks' performance is made by comparing their outputs with the data measured by the nose boom[¶] during flight. The networks gave good results within the entire flight envelope considered.

Figure 15 shows a comparison between the Mach number and the static pressure provided by neural networks and the same data measured during a generic flight. The outputs of the networks closely follow the experimental data within the entire portion of the test.

The training on the flight-test data leads to important benefits in the networks' accuracy for both the Mach number and the static pressure. As an example, in Fig. 16, the absolute values of the static-pressure error are plotted for the following cases: 1) neural networks trained on flight data, 2) neural networks trained on wind-tunnel data, and 3) polynomial functions (coefficients tuned on wind-tunnel data).

By tuning the neural networks on the flight data, the maximum error is reduced from 800 to 200 Pa. It is worth noting that the accuracy of the neural networks trained on flight-test data is of the same order of magnitude as the reference data (the nose-boom accuracy on static pressure for the maximum value of the dynamic pressure during this test is equal to approximately 250 Pa). Comparable results were found within the entire set of flight data examined.

¶For the static pressure, the accuracy of reference data (nose boom) is 3% of the dynamic pressure. Otherwise, the total pressure measured by the nose boom is very precise within the range of angle of attack considered in these flights (below 25 deg), and the error was disregarded.

Table 3 Maximum and mean errors and standard deviation

	Max error	Mean error	Standard deviation
Static pressure, Pa			
Neural networks	200 (0.2%)	58	23
Polynomial functions	240 (0.3%)	94	40
Mach number			
Neural networks	0.0024 (0.7 %)	6.5×10^{-4}	4.1×10^{-4}
Polynomial functions	0.0052 (1.5%)	9.4×10^{-4}	9.4×10^{-4}

VII. Conclusions

Neural network architectures were developed for the computation of Mach number and static pressure during flight. Such an approach demonstrated to be a valid alternative to the use of algorithms based on polynomial calibration functions. The performances of the two approaches are comparable when the neural networks are trained on the same data used to tune the polynomial calibration functions, even if the neural networks are a little more accurate.

The number of coefficients to be stored in the FCCs is larger for the neural networks (approximately 4 times). However, such a comparison is relevant to the cruise-aircraft configuration only, and the difference will probably decrease if deflections of leading-edge and trailing-edge flaps are taken into account. Actually, the compensation of such configuration effects leads to additional polynomial functions. On the contrary, the neural networks could take configuration effects into account by adding new inputs and by modifying the neural architecture, with a limited increase in the number of neural networks coefficients.

One of the advantages of the approach based on the neural networks is that it is possible to use a single network to cover the

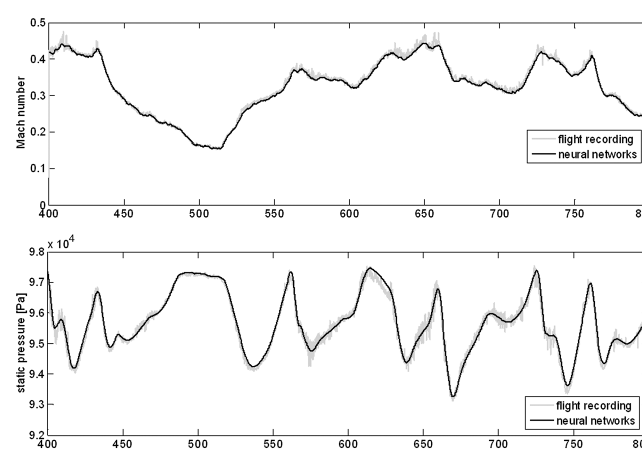

Fig. 15 Mach number and static-pressure neural reconstruction during a flight test.

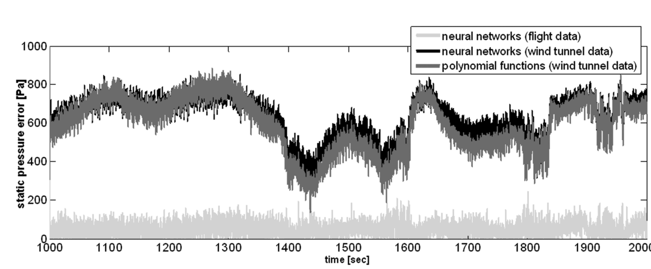

Fig. 16 Static-pressure error during a flight test.

entire flight envelope, and so the fading problems are eliminated and the algorithms architecture is simpler.

In addition, the neural networks show dramatic advantages in terms of time to spent to tune the system on new data. In fact, the tuning of the polynomial functions involves preliminary and partially manual analyses to determine how to manage the data for the polynomial interpolation in multidimensional space. On the contrary, the neural networks can manage a large number of input signals, and consolidated algorithms exist for a fully automatic training. The reduction of calibration time is of particular importance for a high-performance aircraft because the tuning of the air data system is carried out many times during the flight tests of the prototypes, following the gradual enlargement of the flight envelope. In addition, each new aircraft configuration (for example, due to the installation of new external loads) needs a new calibration of the air data system.

References

[1] Menzies, M. A., "Integrated Air Data Sensors," *The Aeronautical Journal*, Vol. 105, Apr. 2001, pp. 223–229.

[2] Lawford, J. A., and Nippress, K. R., "Calibration of Air-Data Systems and Flow Direction Sensors," AGARDograph 300 Flight Test Techniques Series, Vol. 1, AGARD, Rept. AGARD-AG-300, Neuilly-sur-Seine, France, 1988.

[3] Cervia, F., Denti, E., Galatolo, R., and Schettini, F., "Air Data Computation in Fly-by-Wire Flight Control Systems," *Journal of Aircraft*, Vol. 43, No. 2, Mar.–Apr. 2006, pp. 450–455. doi:10.2514/1.16270

[4] Calia, A., Denti, E., Galatolo, R., and Schettini, F., "Fault—Tolerant Procedures for Air Data Elaboration," 25th Congress of International Council of the Aeronautical Sciences, Hamburg Germany, International Council of the Aeronautical Sciences Paper 2006-7.9.4, Sept. 2006.

[5] Calia, A., Poggi, P., and Schettini, F., "Air Data Failure Management in a Full-Authority Fly-by-Wire Control System," *Proceedings of the 2006 IEEE International Conference on Control Applications*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Oct. 2006, pp. 3277–3281.

[6] Gurney, K., *An Introduction to Neural Networks*, UCL Press, London, 1997.

[7] Demuth, H., and Beale, M., *Neural Network Toolbox, for Use with MATLAB: User's Guide*, The MathWorks, Inc., Natick, MA, July 2002.

[8] Rohloff, T. J., Whitmore, S. A., and Catton, I., "Air Data Sensing from Surface Pressure Measurements Using a Neural Network Method," *AIAA Journal*, Vol. 36, No. 11, Nov. 1998, pp. 2094–2101.